Private Information Retrieval from Decentralized Uncoded Caching Databases

Yi-Peng Wei Batuhan Arasli Karim Banawan Sennur Ulukus
Department of Electrical and Computer Engineering
University of Maryland, College Park, MD 20742
ypwei@umd.edu barasli@umd.edu kbanawan@umd.edu ulukus@umd.edu

***Abstract*—We consider the private information retrieval (PIR) problem from decentralized uncoded caching databases. There are two phases in our problem setting, a caching phase, and a retrieval phase. In the caching phase, a data center containing all the K files, where each file is of size L bits, and several databases with storage size constraint μKL bits exist in the system. Each database independently chooses μKL bits out of the total KL bits from the data center to cache through the same probability distribution in a decentralized manner. In the retrieval phase, a user (retriever) accesses N databases in addition to the data center, and wishes to retrieve a desired file privately. We characterize the optimal normalized download cost to be $\frac{D}{L}=\sum_{n=1}^{N+1}\binom{N}{n-1}\mu^{n-1}(1-\mu)^{N+1-n}\left(1+\frac{1}{n}+\cdots+\frac{1}{n^{K-1}}\right)$. We show that uniform and random caching scheme which is originally proposed for decentralized coded caching by Maddah-Ali and Niesen, along with Sun and Jafar retrieval scheme which is originally proposed for PIR from replicated databases surprisingly result in the lowest normalized download cost. This is the decentralized counterpart of the recent result of Attia, Kumar and Tandon for the centralized case.**

I. Introduction

Private information retrieval (PIR) refers to the problem of downloading a desired file from distributed databases while keeping the identity of the desired file private against the databases. In the classical setting of PIR (see Fig. 1), there are N non-communicating databases, each storing the same set of K files. The user wishes to download one of these K files without letting the databases know the identity of the desired file. A simple but highly inefficient way is to download all the files from a particular database, which results in the normalized download cost of $\frac{D}{L}=K$, where L is the file size and D is the total number of downloaded bits from the N databases. The PIR problem originated in the computer science community [1] and has drawn attention in the information theory society with early examples [2]–[5]. Recently, Sun and Jafar [6] have characterized the optimal normalized download cost for the classical PIR problem to be $\frac{D}{L}=\left(1+\frac{1}{N}+\cdots+\frac{1}{N^{K-1}}\right)$. After [6], many interesting variants of PIR have been investigated in [7]–[32]. Most of these previous works consider the case of replicated databases where each database stores the same set of K files.

Coded caching is the problem of placing files in users' local storage caches and designing efficient delivery schemes such that the traffic during the delivery phase is minimized. In the original setup [33] (see Fig. 1), a server with K files connects to N users through an error-free shared link, where each user has a local memory which can store up to M files. The server can arrange the content in each user's local memory in an optimized manner, which is called *centralized coded caching*. Reference [33] proposes a symmetric batch caching scheme, which is shown to be optimal for the case of centralized uncoded placement in [34]. If the set of users in placement and delivery phases varies, the server cannot arrange the files in a centralized manner. Instead, the server treats each user identically and independently which is called *decentralized coded caching* [35]. Reference [35] proposes a uniform and random caching scheme, which is shown to be optimal for the case of decentralized uncoded placement in [34].

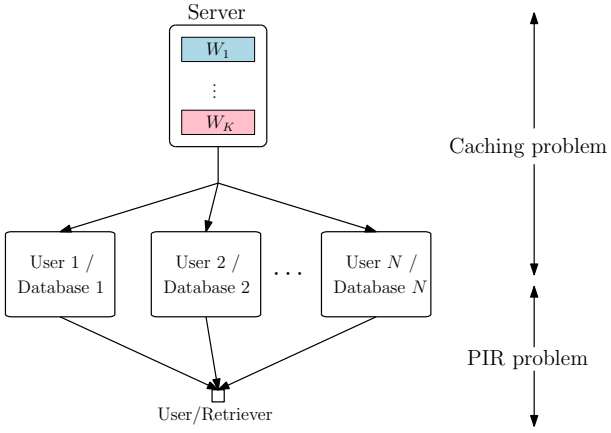

Fig. 1. Joint centralized caching and PIR problem.

The references that are closely related to our work here are [25], [26]. References [25], [26] optimize the content of each database to minimize the download cost. In their problem setting, there is a data center (server) containing all the K files where each file is of size L bits, and the system operates in two phases. In the caching phase, there are N databases in the system with a common storage size constraint μ, i.e., each database can at most store μKL bits, $\frac{1}{N}\leq\mu\leq 1$. In the retrieval phase, a user (retriever) accesses the N databases, and wishes to download a desired file privately. They focus on the *centralized uncoded* caching case, i.e., the set of users in the two phases are identical, and each database stores μKL bits out of the total KL bits. Surprisingly, they show that the symmetric batch caching scheme proposed in [33] results in

This work was supported by NSF Grants CNS 15-26608, CCF 17-13977 and ECCS 18-07348.

the lowest normalized download cost in the retrieval phase.

We consider PIR from *decentralized uncoded* caching databases. Different from [25], [26], the data center does not know in advance which databases the user (retriever) can access in the retrieval phase. This motivates the *decentralized* setting for the caching phase, i.e., each database chooses a subset of bits to store independently according to the same probability distribution. Here, we aim at designing the optimal probability distribution in the caching phase and PIR scheme in the retrieval phase such that the normalized download cost in the retrieval phase is minimized. Another main difference between our work and [25], [26] is that, in the caching phase, [25], [26] require that the N databases altogether can reconstruct the entire K files. In the decentralized setting, where cache placement is probabilistic, we cannot guarantee that any given N databases contain all the bits that exist in the data center. Thus, we allow the user (retriever) access the data center as well as the databases in the retrieval phase.

In this work, we show that uniform and random caching scheme, originally proposed in [35] for decentralized coded caching, results in the lowest expected normalized download cost in the retrieval phase. For the achievability, we apply the PIR scheme in [6] successively for all resulting subfile parts. For the converse, we first apply the lower bound derived in [26], which replaces the random variables for queries and answering strings by the content of the distributed databases in a novel manner extending the lower bounding techniques in [6, Lemma 5 and Lemma 6]. To compare different probability distributions in the caching phase, we focus on the marginal distributions on each separate bit. Then, by using the nature of decentralization and uncoded caching, we further lower bound the normalized download cost. Finally, we show the matching converse for the expected normalized download cost to be $\frac{D}{L}=\sum_{n=1}^{N+1}\binom{N}{n-1}\mu^{n-1}(1-\mu)^{N+1-n}\left(1+\frac{1}{n}+\cdots+\frac{1}{n^{K-1}}\right)$, which yields an exact capacity result for the problem.

II. System Model

We consider a system consisting of one data center and several databases. The data center stores K independent files, labeled as $W_1, W_2, \ldots, W_K$, where each file is of size L bits,

$$H(W_1)=\cdots=H(W_K)=L, \tag{1}$$
$$H(W_1,\ldots,W_K)=H(W_1)+\cdots+H(W_K). \tag{2}$$

Each database has a storage capacity of μKL bits, where $0\leq\mu\leq 1$.

The system operates in two phases: In the caching phase, we consider the case of *uncoded* caching. Due to the storage size constraint, each database at most stores μKL bits out of the total KL bits from the data center. Here, we denote the ith database as DB_i and use random variable Z_i to denote the stored content in DB_i. Therefore, the storage size constraint for DB_i is

$$H(Z_i)\leq\mu KL. \tag{3}$$

We consider the *decentralized* setting for the caching phase, i.e., each database chooses a subset of bits to store independently according to the same probability distribution, denoted by P_H. Rigorously, let random variable H_i denote the indices of the stored bits in DB_i. For N databases, the decentralized caching scheme $\mathcal{H}$ can be specified as

$$\mathbb{P}(\mathcal{H}=(H_1,\ldots,H_N))=\prod_{i=1}^{N}P_H(H_i). \tag{4}$$

In the retrieval phase, the user accesses N databases and the data center. We note that we do not know in advance which N databases are available or which N databases the user will have access to. Here, we also assume that in the retrieval phase, the data center and N databases do not communicate with each other (no collusion). To simplify the notation, we use DB_0 to denote the data center, and therefore $Z_0=(W_1,\ldots,W_K)$ since the data center stores all the K files. The user privately generates an index $\theta\in[K]=\{1,\ldots,K\}$, and wishes to retrieve file W_θ such that it is impossible for either the data center or any individual database to identify θ. For random variables θ, and $W_1,\ldots,W_K$, we have

$$H(\theta,W_1,\ldots,W_K)=H(\theta)+H(W_1)+\ldots+H(W_K) \tag{5}$$

In order to retrieve file W_θ, the user sends $N+1$ queries $Q_0^{[\theta]},\ldots,Q_N^{[\theta]}$ to $\text{DB}_0,\ldots,\text{DB}_N$, where $Q_n^{[\theta]}$ is the query sent to DB_n for file W_θ. Note that the queries are independent of the realization of the K files. Therefore,

$$I(W_1,\ldots,W_K;Q_0^{[\theta]},\ldots,Q_N^{[\theta]})=0. \tag{6}$$

Upon receiving the query $Q_n^{[\theta]}$, DB_n replies with an answering string $A_n^{[\theta]}$, which is a function of $Q_n^{[\theta]}$ and Z_n. Therefore, $\forall\theta\in[K],\forall n\in\{0\}\cup[N]$, $H(A_n^{[\theta]}|Q_n^{[\theta]},Z_n)=0$.

After receiving the answering strings $A_0^{[\theta]},\ldots,A_N^{[\theta]}$ from $\text{DB}_0,\ldots,\text{DB}_N$, the user needs to decode the desired file W_θ reliably. By using Fano's inequality, we have the following reliability constraint

$$H\left(W_\theta|Q_0^{[\theta]},\ldots,Q_N^{[\theta]},A_0^{[\theta]},\ldots,A_N^{[\theta]}\right)=o(L), \tag{7}$$

where $o(L)$ denotes a function such that $\frac{o(L)}{L}\to 0$ as $L\to\infty$.

To ensure that individual databases do not know which file is retrieved, we have the following privacy constraint, $\forall n\in\{0\}\cup[N]$, $\forall\theta\in[K]$,

$$(Q_n^{[1]},A_n^{[1]},W_1,\ldots,W_K)\sim(Q_n^{[\theta]},A_n^{[\theta]},W_1,\ldots,W_K), \tag{8}$$

where $A\sim B$ means that A and B are identically distributed.

Given that each file is of size L bits, for a fixed K, μ and decentralized caching probability distribution P_H, let $\mathcal{H}$ denote the indices of the cached bits in the N databases available in the retrieval phase. The probability distribution of $\mathcal{H}$ is specified in (4). Let $D_{\mathcal{H}}^{[\theta]}$ represent the number of downloaded bits via the answering strings $A_{0:N}^{[\theta]}$, where $A_{0:N}^{[\theta]}=(A_0^{[\theta]},\ldots,A_N^{[\theta]})$. Then, $D_{\mathcal{H}}^{[\theta]}=\sum_{n=0}^{N}H\left(A_n^{[\theta]}\right)$. We further denote $D_{\mathcal{H}}$ as the expected number of downloaded bits with respect to different file requests, i.e., $D_{\mathcal{H}}=E_\theta\left[D_{\mathcal{H}}^{[\theta]}\right]$.

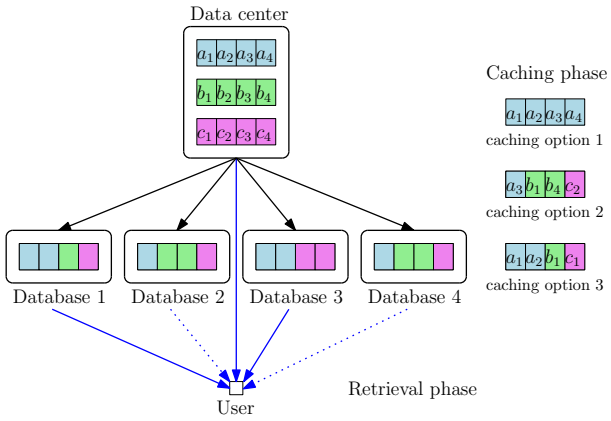

Fig. 2. PIR from decentralized databases with $K = 3$, $N = 2$, and $\mu = \frac{1}{3}$.

Finally, we denote D as the expected number of downloaded bits with respect to different realizations of the cached bit indices, i.e., $D = E_{\mathcal{H}}[D_{\mathcal{H}}]$. A pair (D, L) is achievable if there exists a PIR scheme satisfying the reliability constraint (7) and the privacy constraint (8). The optimal normalized download cost D^* is defined as

$$D^* = \inf\left\{\frac{D}{L} : (D, L) \text{ is achievable}\right\}. \tag{9}$$

In this work, we aim at characterizing the optimal normalized download cost and finding the optimal decentralized caching probability distribution.

Next, we illustrate the system model and the problem considered with a simple example of $K = 3$ files and $N = 2$ databases in the retrieval phase; see Fig. 2. Consider a data center storing $K = 3$ files where each file is of size 4 bits. In the caching phase, there are 4 databases in the system, and each database can at most store 4 bits. Each database can always store the first file, which is of size 4 bits, as caching option 1 in Fig. 2. Or each database can uniformly and randomly choose 4 bits out of total 12 bits from the data center to store. One of the realizations is shown as caching option 2 in Fig. 2. Each database can also choose 2 bits from the first file and 1 bit each from the remaining two files to store, where one of the realizations is shown as caching option 3 in Fig. 2. We require each database to use the same probability distribution to choose the bits to store in order to satisfy the decentralized requirement. In this example, we assume that the user can access the data center and $N = 2$ databases in the retrieval phase, say the first and the third database, and the user wishes to download a file privately.

III. Main Results

We characterize the optimal normalized download cost for PIR from decentralized uncoded caching databases in the following theorem.

Theorem 1 *For PIR from decentralized uncoded caching databases with K files, where each file is of size L bits, N databases in addition to a data center available in the retrieval phase, and a storage size constraint μKL, $0 < \mu < 1$, bits for each database, the optimal normalized download cost is*

$$\frac{D}{L} = \sum_{n=1}^{N+1} \binom{N}{n-1} \mu^{n-1}(1-\mu)^{N+1-n} \times \left(1 + \frac{1}{n} + \cdots + \frac{1}{n^{K-1}}\right). \tag{10}$$

The achievability scheme is provided in [36, Sec. 4], and the converse proof is shown in [36, Sec. 5] due to space limitations here. The essence of the achievability and the converse proofs are captured in the following representative example, which deals with $K = 3$, $N = 2$ and shows the main ingredients of Theorem 1 without loss of generality.

A. Representative Example: $K = 3$ and $N = 2$

In this example, we consider the case where the data center stores $K = 3$ independent files labeled as A, B, and C, where each file is of size L bits. In the caching phase, several databases with storage capacity of $3\mu L$ bits are present in the system. We will show that the optimal normalized download cost is $\frac{D}{L} = \frac{17}{18}\mu^2 - \frac{5}{2}\mu + 3$ when $N = 2$ databases in addition to the data center are available in the retrieval phase.

1) Achievability Scheme: In the caching phase, to satisfy the storage size constraint, each database randomly and uniformly stores $3\mu L$ bits out of total $3L$ bits from the data center. Each database operates independently through the same probability distribution resulting in decentralized caching.

In the retrieval phase, suppose $N = 2$ databases, labeled as DB_1 and DB_2, in addition to the data center, labeled as DB_0, are available to the user, and the user wishes to retrieve file A privately. Let us first focus on one file, say A. We can partition file A into four subfiles

$$A = (A_0, A_{0,1}, A_{0,2}, A_{0,1,2}), \tag{11}$$

where, for $S \subseteq \{0, 1, 2\}$, A_S denotes the bits of file A which are stored in databases in S. For example, A_0 denotes the bits of file A only stored in DB_0 and $A_{0,2}$ denotes the bits of file A stored in DB_0 and DB_2 and so on. Since each bit is stored in the data center, 0 exists in the label of every partition. By the law of large numbers,

$$|A_S| = L\mu^{|S|-1}(1-\mu)^{3-|S|} + o(L), \tag{12}$$

when the file size is large enough. We can do the same partitions for files B and C.

To retrieve file A privately, we first retrieve the subfile $A_{0,1,2}$ privately. We apply the PIR scheme proposed in [6] to retrieve the subfile $A_{0,1,2}$. Subfile $A_{0,1,2}$ is replicated in 3 databases and the total number of files is 3 since we also have $B_{0,1,2}$ and $C_{0,1,2}$. Therefore, we download

$$L\mu^2\left(1 + \frac{1}{3} + \frac{1}{9}\right) + o(L) \tag{13}$$

bits. We also need to retrieve the subfile $A_{0,1}$ privately. Subfile $A_{0,1}$ is replicated in 2 databases and the total number of files is 3 since we also have $B_{0,1}$ and $C_{0,1}$. By applying the PIR

scheme in [6], we download

$$L\mu(1-\mu)\left(1+\frac{1}{2}+\frac{1}{4}\right)+o(L) \tag{14}$$

bits. Next, we retrieve $A_{0,2}$ privately. Using [6], we download

$$L\mu(1-\mu)\left(1+\frac{1}{2}+\frac{1}{4}\right)+o(L) \tag{15}$$

bits. Finally, we retrieve A_0 privately. Using [6], we download

$$L(1-\mu)^2(1+1+1)+o(L) \tag{16}$$

bits. By adding (13), (14), (15) and (16), we show that the normalized download cost $\frac{17}{18}\mu^2-\frac{5}{2}\mu+3$ is achievable.

2) Converse Proof: Here, we show that among all the decentralized caching probability distributions P_H, the lowest normalized download cost for $N=2$ databases matches the achievablility. Given a decentralized caching probability distribution P_H, we have a resulting $\mathcal{H}$ in the retrieval phase.

We lower bound $D_{\mathcal{H}}$ first. In the retrieval phase, the stored content of DB_0, DB_1, and DB_2 are fixed and uncoded, i.e., Z_0, Z_1 and Z_2 are fixed and uncoded. We apply the lower bound in [26, Eqn. (31)] as the lower bound for $D_{\mathcal{H}}$. Therefore,

$$\begin{aligned}
D_{\mathcal{H}} \geq\, & L+\frac{4}{27}\sum_{k=1}^{3}H(W_k)+\frac{11}{108}\sum_{i=0}^{2}\sum_{k=1}^{3}H(W_k|Z_i)\\
&+\frac{17}{54}\sum_{i=0}^{2}\sum_{k=1}^{3}H(W_k|Z_{[0:2]\setminus i})+o(L) && (17)\\
=\, & \frac{13}{9}L+\frac{11}{108}\sum_{i=1}^{2}\sum_{k=1}^{3}H(W_k|Z_i)\\
&+\frac{17}{54}\sum_{k=1}^{3}H(W_k|Z_1,Z_2)+o(L) && (18)\\
\geq\, & \frac{13}{9}L+\frac{11}{108}\left(3L-3\mu L+3L-3\mu L\right)\\
&+\frac{17}{54}\sum_{k=1}^{3}H(W_k|Z_1,Z_2)+o(L) && (19)\\
=\, & \frac{37}{18}L-\frac{11}{18}\mu L+\frac{17}{54}H(W_{1:3}|Z_1,Z_2)+o(L), && (20)
\end{aligned}$$

where (18) holds due to $Z_0=(W_1,W_2,W_3)$, and (19) holds due to (3). We note that different $\mathcal{H}$ results in different Z_1 and Z_2. We lower bound D now. From (20), we have

$$\begin{aligned}
D =& E_{\mathcal{H}}\left[D_{\mathcal{H}}\right]\\
\geq& \frac{37}{18}L-\frac{11}{18}\mu L+\frac{17}{54}E_{\mathcal{H}}\left[H(W_{1:3}|Z_1,Z_2)\right]+o(L). \quad (21)
\end{aligned}$$

Let random variables $X_{i,j}^{(n)}$, $i=1,\ldots,L$, $j=1,\ldots,K$, be the indicator functions showing that the ith bit of file W_j is cached in DB_n or not, i.e., $X_{i,j}^{(n)}=1$ means that the ith bit of file W_j is stored in DB_n and $X_{i,j}^{(n)}=0$ means that it is not stored in DB_n. For DB_1 we have

$$X_{1,1}^{(1)}+\cdots+X_{L,1}^{(1)}+\cdots+X_{1,3}^{(1)}+\cdots+X_{L,3}^{(1)}\leq 3\mu L \tag{22}$$

due to the storage size constraint in (3). We note that P_H induces probability measures on random variables $X_{i,j}^{(n)}$, and let $X_{i,j}^{(n)}=1$ with probability $p_{i,j}$, where we remove the superscript n since each database adopts the same probability distribution P_H to choose the cached bits due to the decentralized property. By taking expectation on (22) and applying the linearity of expectation, we have $E[X_{1,1}^{(1)}]+\cdots+E[X_{L,3}^{(1)}]\leq 3\mu L$, which yields

$$p_{1,1}+\cdots+p_{L,3}\leq 3\mu L. \tag{23}$$

Let random variables $V_{i,j}$, $i=1,\ldots,L$, $j=1,\ldots,K$, be the indicator functions showing that the ith bit of file W_j is not cached in DB_1 and DB_2, i.e., $V_{i,j}=1$ means that the ith bit of file W_j is not stored in either DB_1 or DB_2. Therefore,

$$V_{i,j}=(1-X_{i,j}^{(1)})(1-X_{i,j}^{(2)}). \tag{24}$$

Now, we can evaluate $E_{\mathcal{H}}\left[H(W_{1:3}|Z_1,Z_2)\right]$ in (21) as follows

$$\begin{aligned}
&E_{\mathcal{H}}\left[H(W_{1:3}|Z_1,Z_2)\right]\\
&\quad=E[V_{1,1}+\cdots+V_{L,3}] && (25)\\
&\quad=(1-p_{1,1})^2+\cdots+(1-p_{L,3})^2. && (26)
\end{aligned}$$

Therefore, continuing from (21), we have

$$\begin{aligned}
D \geq& \frac{37}{18}L-\frac{11}{18}\mu L+\frac{17}{54}\left[(1-p_{1,1})^2+\cdots+(1-p_{L,3})^2\right]\\
&+o(L), && (27)
\end{aligned}$$

where $p_{1,1},\ \ldots,\ p_{L,3}$ are subject to (23). To further lower bound the right hand side of (27), we minimize the right hand side with respect to $p_{i,j}$ subject to (23). Hence, we consider the following Lagrangian

$$\begin{aligned}
L(p_{1,1},\ldots,p_{L,3},\lambda)=&(1-p_{1,1})^2+\cdots+(1-p_{L,3})^2\\
&+\lambda\left(p_{1,1}+\cdots+p_{L,3}-3\mu L\right). \quad (28)
\end{aligned}$$

From the KKT conditions, we have

$$\lambda=2(1-p_{i,j}),\quad i=1,\ldots,L,\quad j=1,2,3. \tag{29}$$

Thus, we can further lower bound (27) by letting $p_{1,1}=\cdots=p_{L,3}=\mu$, and we have

$$\begin{aligned}
\frac{D}{L}\geq\, & \frac{37}{18}-\frac{11}{18}\mu+\frac{17}{54}\left[3(1-\mu)^2\right]+\frac{o(L)}{L} && (30)\\
=\, & \frac{17}{18}\mu^2-\frac{5}{2}\mu+3+\frac{o(L)}{L}. && (31)
\end{aligned}$$

Therefore, we show that the optimal normalized download cost is $\frac{17}{18}\mu^2-\frac{5}{2}\mu+3$ when $N=2$ databases in addition to the data center are available in the retrieval phase. To achieve the optimal normalized download cost, each database should randomly and uniformly store the bits in the caching phase.

IV. Conclusion and Discussions

We considered the PIR problem from decentralized uncoded caching databases. We showed that uniform and random decentralized caching scheme, originally proposed in [35] for the problem of decentralized coded caching, results in the

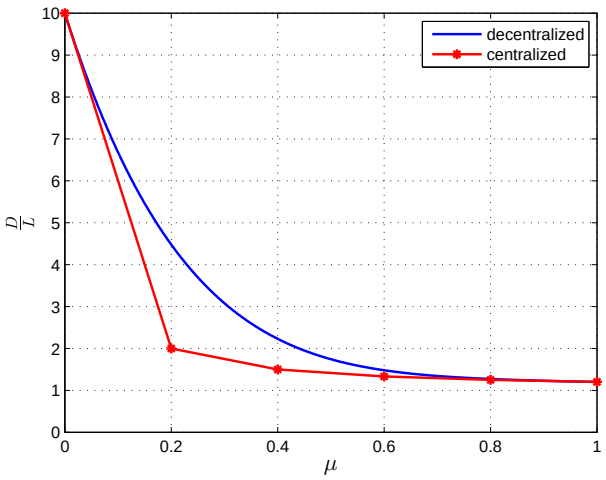

Fig. 3. Comparison between centralized and decentralized settings.

lowest expected normalized download cost in the PIR phase.

Finally, we compare the optimal expected download cost of our problem and the optimal download cost in the centralized setting in [25], [26]. For a fair comparison, we allow the user to have access also to the data center in addition to the N databases in the centralized setting as well. This is different from the problem setting in [25], [26]. Nevertheless, we can show [36] that symmetric batch caching scheme is still optimal for this extended problem setting where the data center also participates in the PIR stage. Rigorously, the optimal trade-off between storage and download cost in this case is given by the lower convex envelope of the following $(\mu, D(\mu))$ pairs, for $t = 0, 1, \ldots, N$,

$$\left(\mu = \frac{t}{N}, D(\mu) = \sum_{k=0}^{K-1} \frac{1}{(t+1)^k}\right). \tag{32}$$

We plot the resulting curves for centralized and decentralized cases in Fig. 3 for $K = 10$ and $N = 5$.

REFERENCES

[1] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan. Private information retrieval. *Journal of the ACM*, 45(6):965–981, 1998.

[2] N. B. Shah, K. V. Rashmi, and K. Ramchandran. One extra bit of download ensures perfectly private information retrieval. In *IEEE ISIT*, June 2014.

[3] T. Chan, S. Ho, and H. Yamamoto. Private information retrieval for coded storage. In *IEEE ISIT*, June 2015.

[4] A. Fazeli, A. Vardy, and E. Yaakobi. Codes for distributed PIR with low storage overhead. In *IEEE ISIT*, June 2015.

[5] R. Tajeddine and S. El Rouayheb. Private information retrieval from MDS coded data in distributed storage systems. In *IEEE ISIT*, July 2016.

[6] H. Sun and S. A. Jafar. The capacity of private information retrieval. *IEEE Transactions on Information Theory*, 63(7):4075–4088, July 2017.

[7] H. Sun and S. A. Jafar. The capacity of robust private information retrieval with colluding databases. *IEEE Transactions on Information Theory*, 64(4):2361–2370, April 2018.

[8] H. Sun and S. A. Jafar. The capacity of symmetric private information retrieval. *IEEE Trans. on Info. Theory*, 65(1):322–329, January 2019.

[9] K. Banawan and S. Ulukus. The capacity of private information retrieval from coded databases. *IEEE Transactions on Information Theory*, 64(3):1945–1956, March 2018.

[10] Q. Wang and M. Skoglund. Symmetric private information retrieval for MDS coded distributed storage. 2016. Available at arXiv:1610.04530.

[11] R. Freij-Hollanti, O. Gnilke, C. Hollanti, and D. Karpuk. Private information retrieval from coded databases with colluding servers. *SIAM Journal on Applied Algebra and Geometry*, 1(1):647–664, 2017.

[12] H. Sun and S. A. Jafar. Private information retrieval from MDS coded data with colluding servers: Settling a conjecture by Freij-Hollanti et al. *IEEE Trans. on Info. Theory*, 64(2):1000–1022, February 2018.

[13] K. Banawan and S. Ulukus. Multi-message private information retrieval: Capacity results and near-optimal schemes. *IEEE Transactions on Information Theory*, 64(10):6842–6862, October 2018.

[14] K. Banawan and S. Ulukus. The capacity of private information retrieval from byzantine and colluding databases. *IEEE Trans. on Info. Theory*, 65(2):1206–1219, February 2019.

[15] R. Tandon. The capacity of cache aided private information retrieval. In *Allerton Conference*, September 2017.

[16] S. Kadhe, B. Garcia, A. Heidarzadeh, S. El Rouayheb, and A. Sprintson. Private information retrieval with side information. 2017. Available at arXiv:1709.00112.

[17] Y.-P. Wei, K. Banawan, and S. Ulukus. Fundamental limits of cache-aided private information retrieval with unknown and uncoded prefetching. *IEEE Trans. on Info. Theory*, 65(5):3215–3232, May 2019.

[18] Y.-P. Wei, K. Banawan, and S. Ulukus. Cache-aided private information retrieval with partially known uncoded prefetching: Fundamental limits. *IEEE JSAC*, 36(6):1126–1139, June 2018.

[19] Z. Chen, Z. Wang, and S. A. Jafar. The capacity of private information retrieval with private side information. 2017. Available at arXiv:1709.03022.

[20] Y.-P. Wei, K. Banawan, and S. Ulukus. The capacity of private information retrieval with partially known private side information. 2017. Available at arXiv:1710.00809.

[21] H. Sun and S. A. Jafar. The capacity of private computation. 2017. Available at arXiv:1710.11098.

[22] K. Banawan and S. Ulukus. Asymmetry hurts: Private information retrieval under asymmetric traffic constraints. 2018. Available at arXiv:1801.03079.

[23] K. Banawan and S. Ulukus. Private information retrieval through wiretap channel II: Privacy meets security. 2018. Available at arXiv:1801.06171.

[24] K. Banawan and S. Ulukus. Noisy private information retrieval: On separability of channel coding and information retrieval. 2018. Available at arXiv:1807.05997.

[25] M. Abdul-Wahid, F. Almoualem, D. Kumar, and R. Tandon. Private information retrieval from storage constrained databases – coded caching meets PIR. 2017. Available at arXiv:1711.05244.

[26] M. A. Attia, D. Kumar, and R. Tandon. The capacity of private information retrieval from uncoded storage constrained databases. 2018. Available at arXiv:1805.04104.

[27] Y.-P. Wei and S. Ulukus. The capacity of private information retrieval with private side information under storage constraints. 2018. Available at arXiv:1806.01253.

[28] C. Tian, H. Sun, and J. Chen. Capacity-achieving private information retrieval codes with optimal message size and upload cost. 2018. Available at arXiv:1808.07536.

[29] S. Kumar, A. G. i Amat, E. Rosnes, and L. Senigagliesi. Private information retrieval from a cellular network with caching at the edge. 2018. Available at arXiv:1809.00872.

[30] S. Li and M. Gastpar. Converse for multi-server single-message PIR with side information. 2018. Available at arXiv:1809.09861.

[31] R. Tajeddine, A. Wachter-Zeh, and C. Hollanti. Private information retrieval over networks. 2018. Available at arXiv:1810.08941.

[32] K. Banawan, B. Arasli, Y.-P. Wei, and S. Ulukus. The capacity of private information retrieval from heterogeneous uncoded caching databases. *IEEE Trans. on Info. Theory*. Submitted February 2019. Also available at arXiv:1902.09512.

[33] M. A. Maddah-Ali and U. Niesen. Fundamental limits of caching. *IEEE Transactions on Information Theory*, 60(5):2856–2867, May 2014.

[34] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr. The exact rate-memory tradeoff for caching with uncoded prefetching. *IEEE Transactions on Information Theory*, 64(2):1281–1296, February 2018.

[35] M. A. Maddah-Ali and U. Niesen. Decentralized coded caching attains order-optimal memory-rate tradeoff. *IEEE/ACM Transactions on Networking*, 23(4):1029–1040, August 2015.

[36] Y.-P. Wei, B. Arasli, K. Banawan, and S. Ulukus. The capacity of private information retrieval from decentralized uncoded caching databases. 2018. Available at arXiv:1811.11160.